INFORMS Journal on Computing
Vol. 9, No. 1, Winter 1997

0899-1499/97/0901-0100 $05.00

Art Gallery Problems for Convex Nested Polygons

J. BHADURY / *Faculty of Administration, University of New Brunswick, Fredericton, Canada, Email: bhadury@unb.ca*

V. CHANDRU / *Department of Computer Science, Indian Institute of Science, Bangalore, India, Email: chandru@csa.iisc.ernet.in*

A. MAHESHWARI / *Department of Computer Science, Carleton University, Ottawa, Canada, Email: maheshwa@chaos.scs.carleton.ca*

R. CHANDRASEKARAN / *School of Management, The University of Texas at Dallas, Richardson, Texas, USA, Email: rchandra@utdallas.edu*

(Received: September 1995; accepted December 1995)

In this article, we study a class of Art Gallery problems that are defined on a pair of convex nested polygons. Polynomial time algorithms are presented for all these problems, by reducing them to the Circle Covering problem, or by relating them to the Minimal Nested Polygon problem. Then it is shown that these problems can also be solved in polynomial time by formulating them as Integer Programs with the *Circular Ones* property. Finally the paper discusses how these algorithms can be efficiently implemented in parallel.

Art Gallery Problems refer to the general class of problems in Computational Geometry where it is required to watch an entire polygon by placing guards. One of the earliest references to such problems was by Victor Klee (O'Rourke[25]), who in 1973 posed the problem of determining the minimum number of guards that are necessary to watch the interior of an n-sided polygon. Soon after, Chvátal[9] showed that $\lfloor n / 3 \rfloor$ guards are sometimes necessary and always sufficient to watch the entire interior of a polygon with n vertices—a result thus dubbed "Chvátal's Art Gallery Theorem". Since then there has been a plethora of work done on similar problems; for instance, if a polygon has h holes in it, then Aggarwal[1] (p. 110) and Shermer (O'Rourke[25] (p. 128)) showed independently with two counterexamples that $\lfloor (n + h) / 3 \rfloor$ guards may be necessary, where n is the total number of vertices of the polygon and the holes. Later it was shown independently by Hoffman, Kaufmann, and Kriegel[14] and Bjorling-Sachs and Souvaine[6] that $\lfloor (n + h) / 3 \rfloor$ guards are also sufficient. Two sources for the literature in this area are O'Rourke[25] and Shermer.[28]

However, although $\lfloor n / 3 \rfloor$, or $\lfloor (n + h) / 3 \rfloor$ in case of holes, may be sometimes necessary, in many cases fewer guards will suffice. This gives rise to the following optimization problem: Given a simple polygon, what is the minimum number of guards that are necessary to watch its interior? This problem is also referred to as a Minimal Guard Cover problem. Several variants of this problem are possible, depending on the type of the polygon, (whether it has "holes" in its interior or not), the allowable guard positions (whether they are permitted to be placed anywhere in the interior, or only on vertices of the polygon), the type of guards (point and stationary versus mobile guards), etc. Many such Minimal Guard Cover problems have been shown to be NP Hard. For example, O'Rourke and Suppowit[26] have shown that when the polygon has holes and the guards are allowed to be located anywhere in the interior of the polygon, this problem is NP Hard. Lee and Lin[17] showed that the problem remains NP Hard even when the polygon does not have any holes; however, their result is for vertex guards. Aggarwal[1] extended his result to point guards.

In this article, we consider a special class of Art Gallery problems that are defined for nested polygons, i.e., polygons with one hole, and show how these problems can be solved in polynomial time. The basic problem considered here can be described as follows: given a pair of convex nested polygons, P_{in} and P_{out}, with $P_{in} \subseteq P_{out}$, what is the minimum number of guards needed, and their respective positions, in order to watch designated areas if the guards themselves are restricted to be only in some other designated areas? The inner polygon P_{in} is assumed to be opaque, i.e., it is not possible to see through this polygon, and to avoid trivial cases, is assumed to have a positive area. Three designated areas will be considered here—the boundary of P_{in}, the boundary of P_{out} and the complete annulus between P_{in} and P_{out}, (including these two boundaries). This gives rise to nine different problems that will be referred to as shown in Table 1.

Thus according to our nomenclature, Problem (P_{out}/P_{in}) refers to the case where the guards will be required to watch the entire boundary of P_{in}, while restricted to be only on the boundary of P_{out}. It is worthwhile to note that although Art Gallery problems for convex nested polygons have not been studied before in the literature, similar problems have. For example, the sufficiency of $\lfloor (n + h) / 3 \rfloor$ guards was shown for $h = 1$, by Shermer.[27] Further, the annulus between a pair of nested convex polygons is closely related to the interior of a spiral polygon, and there are numerous results on the visibility properties of spiral polygons; see for example, Everett and Corneil,[11] MacDonald and Shermer.[19] Other Art Gallery-type results on spiral polygons include Nilsson and Wood,[22] Vishwanathan,[30] and Liaw, Huang, and Lee.[18]

Subject classifications: Computer science.
Other key words: computational geometry; art gallery problems; convex polygons.

Table I. List of Problems

Guard Positions	Areas Watched: Boundary of P_{in}	Boundary of P_{out}	Annulus Between P_{in} and P_{out}
Boundary of P_{in}	Problem (P_{in}/P_{in})	Problem (P_{in}/P_{out})	Problem (P_{in}/Ann)
Boundary of P_{out}	Problem (P_{out}/P_{in})	Problem (P_{out}/P_{out})	Problem (P_{out}/Ann)
Annulus Between P_{in} and P_{out}	Problem (Ann/P_{in})	Problem (Ann/P_{out})	Problem (Ann/Ann)

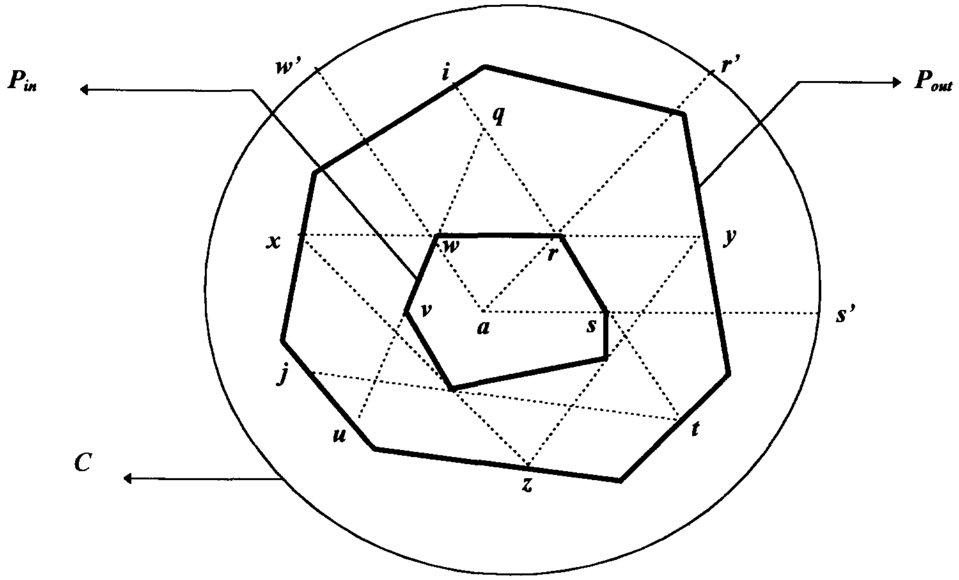

Figure 1. Convex Nested Polygons.

The remainder of the article is divided as follows. In Section 1 we state the preliminaries and then Section 2 presents the algorithms for solving Problems (P_{in}/P_{in}) through (Ann/Ann). Section 3 shows how these problems can also be solved by formulating them as Integer Linear Programs and discusses efficient parallel implementation of the algorithms in Section 2. Finally, Section 4 summarizes the results of this paper.

1. Preliminaries

This section describes the notation and some construction procedures that are used in this paper. In addition, some basic results are also stated and proved. The model of computation used is similar to the one in Aggarwal et al.[2]—the usual random access machine that allows simple arithmetic operations (like +, −, /, *, and taking square roots of real numbers) to be performed in unit time.

It will always be assumed, unless otherwise mentioned, that the two polygons P_{in} and P_{out} are convex and have n and m vertices respectively. The boundaries of these two polygons will be denoted by bd(P_{in}) and bd(P_{out}) respectively, and the entire annulus between P_{in} and P_{out}, including bd(P_{in}) and bd(P_{out}), by (P_{out}–P_{in}). Further, for any two distinct points x, y on bd(P_{in}) or bd(P_{out}), $[x, y]$ will be used to denote the segment of the boundary encountered in a clockwise traversal from x to y. (x, y) will be used to denote the corresponding open segment obtained from $[x, y]$ by excluding the two endpoints x and y.

Consider any point q in the annulus, as shown in Figure 1. Suppose that we wish to find the last point on bd(P_{in}) that is visible to q, in a clockwise traversal of bd(P_{in}). This point is found by drawing the clockwise tangent from q to P_{in}, and choosing the vertex of P_{in} that this tangent intersects (in case the tangent intersects two vertices, the more clockwise of these is chosen), as given by point s in the figure. Let t be the point of intersection of bd(P_{out}) with a ray that originates from q in the direction of s. Then t is the last point on bd(P_{out}) that is visible to q in a clockwise traversal of bd(P_{out}). Let the corresponding last points that are visible to q in a counterclockwise traversal of bd(P_{in}) and bd(P_{out}) be represented by v and u respectively, as shown in the figure. Then, in a clockwise traversal of bd(P_{in}) bd(P_{out})), the entire segment that begins with the point v (u) and ends with s (t), will be referred to as the *Inner Visibility Segment* (*Outer Visibility Segment*) of q. These two segments will be denoted by In(q) and Out(q) respectively. Thus for any point q in the annulus, the two segments In(q) and Out(q) represent portions of the boundary of P_{in} and P_{out} respectively, that are visible from q. As is evident from Figure 1, corresponding to the vertex r of P_{in}, the segment In(r) is given by the two edges $[w, r]$ and $[r, s]$. Similarly, the segment Out(r) consists of the segment $[x, t]$. Since we have assumed that P_{in} is not a point, it can also be concluded that for any point $q \in (P_{out}-P_{in})$, both In(q) and Out(q) have positive lengths, i.e., both segments have a relative interior.

Circle Covering Problem.

This problem is stated as follows: given a circle and a set of arcs that collectively cover it, find the minimum number of arcs that will cover the circle. The algorithm for solving it is given in Lee and Lee.[31] In addition, it is also known that if the arcs are given in a sorted order, this problem can be solved in time that is linear in the number of arcs; see Chandru, Rajan, and Swaminathan[7] for the algorithm.

Minimal Nested Polygon Problem.

This problem is stated as follows: given a pair of nested polygons P_{in} and P_{out}, find a polygon that is contained in $(P_{out}-P_{in})$ and has the minimum number of edges. Throughout this paper, we will assume that for any given configuration of P_{in} and P_{out}, the Minimal Nested Polygon is denoted by P^* and that it has k vertices. For example if P_{in} and P_{out} are as given in Figure 1, P^* is given by the triangle x, y, z and hence $k = 3$. The Minimal Nested Polygon P^* is convex if and only if the convex hull of P_{in} does not properly intersect with bd(P_{out}), as is the case when P_{in} and P_{out} are both convex polygons. It is possible to test for this condition in $O(n + m)$, i.e., linear time; see Preparata and Shamos[24] for the algorithm. If this condition holds, then P^* is guaranteed to be convex and the best available algorithm is given in Aggarwal et al.,[2] the running time of their algorithm is $O((n + m)\log k)$. Since we have assumed in this paper that both P_{in} and P_{out} are convex, unless otherwise specified, P^* is always guaranteed to be convex.

Greedy Structure.

In this paper, we define a *Greedy Structure* for any point $x \in (P_{out}-P_{in})$, that is denoted by G(x). Its construction is best explained with an example; hence, consider the point x in Figure 1. Draw the clockwise tangent from x to P_{in}, and let y be the point where this tangent intersects bd(P_{out}), as shown in the figure. Then y and the line segment $[x, y]$ are considered the first vertex and the first edge respectively of this greedy structure of x. The next vertices and edges are obtained by repeating this procedure from the point y and drawing successive clockwise tangents to P_{in} until a point is reached, where x becomes visible for the first time; this point is given by z in the figure. Once this happens, the last edge of the greedy structure of x is given by drawing one more clockwise tangent from this point to P_{in} and the point of intersection of this last clockwise tangent with bd(P_{out}) is regarded as the last vertex of this greedy structure. If the procedure is repeated by taking counterclockwise tangents from x, then the resulting structure is referred to as the *counterclockwise* greedy structure of x and denoted by $G_a(x)$. For some point $x \in$ bd(P_{out}), it may be possible that the last (counterclockwise) clockwise tangent in ($G_a(x)$) G(x) intersects bd(P_{out}) at x itself; in such cases, x will be referred to as a *tight* point. For example, the point x shown in Figure 1 is a tight point and G(x) and $G_a(x)$ are given by the triangle x, y, z. From this construction procedure, it should also be evident that given a point x, both $G_a(x)$ and G(x) can be constructed in $O(n + m)$ time.

Circle Mapping Transformation.

This transformation will be frequently used in the article to find the optimal placement of guards. Given any convex polygon and a set of segments on its boundary, this transformation maps this polygon onto a circle, and these segments onto corresponding arcs on the circumference of the circle. To illustrate this transformation, assume that the polygon to be transformed is given by P_{in} in Figure 1, with the edges $[w, r]$ and $[r, s]$ being two segments on its boundary. The transformation first chooses a point inside the polygon, as given by a in the figure. Then a circle is drawn, with a as the center, and a radius that is large enough to ensure that P_{in} is a proper subset of this circle. This circle, which is given by C in the figure, is assumed to be the mapping of the polygon under this transformation. After this, every segment on the boundary of P is mapped onto the circle as follows: the two endpoints of the segment are first chosen, and two rays are drawn from a in the directions of these two endpoints. The two points of intersection of these two rays with the circle C are defined as the images of these two endpoints under this transformation. Since these images occur on a circle, they define an arc; this arc is considered the mapping of this segment onto this circle under the transformation. For example in Figure 1, the segments $[w, r]$ and $[r, s]$ get mapped onto arcs $[w', r']$ and $[r', s']$ on C respectively. Finding the two points of intersection of the two rays with the circle will require computing the square root of a real number; however, under our assumed model of computation this can be done in constant time. Therefore it can be concluded that this transformation can be performed in time that is linear in the number of vertices of the given convex polygon, and the number of segments on its boundary.

We begin by showing that there is a close relationship between the Minimal Nested Polygon Problem and Problem (P_{out}/P_{out}), and this relationship holds even when the two polygons are not convex.

Lemma 1. *Given a pair of nested polygons P_{in} and P_{out}, that are not necessarily convex, it can be assumed that any optimal solution to Problem (P_{out}/P_{out}) for this pair of polygons has at least $\lceil k / 2 \rceil$ guards.*

Proof. Suppose that the optimal solution to Problem (P_{out}/P_{out}) for this pair of nested polygons has p guards, where $p < \lceil k / 2 \rceil$. Then we can construct a nested polygon that is completely contained in the annulus $(P_{out}-P_{in})$, and has less than k vertices, which contradicts the optimality of P^*.

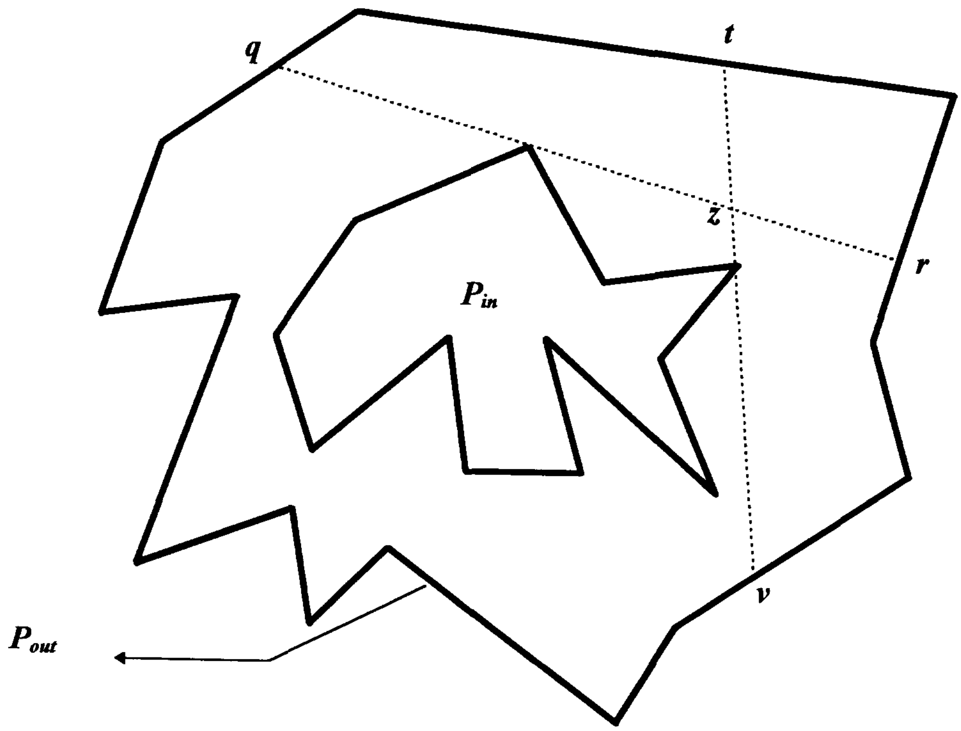

Figure 2. Constructing the Optimal Nested Polygon.

The polygon is constructed in the following manner. From the optimal solution, choose any guard position, say q, as shown in Figure 2. In a clockwise traversal of bd(P_{out}) that begins at q, let r be the last point on bd(P_{out}) that is visible to q. Then if we continue this clockwise traversal beyond r, there must be a contiguous set of points on bd(P_{out}) encountered in the immediate neighborhood of r, that cannot be watched by q and hence, must be watched by some other guard, say t. Condition now on two separate cases:

Case (i). $t \in [q, r)$, as shown in Figure 2. In a clockwise traversal of bd(P_{out}) that begins at t, let v be the last point on bd(P_{out}) that is visible to t. Then we can claim that since t watches the contiguous set of points mentioned above, $v \in (r, q]$ in this case.

Case (ii). $t \in [r, q]$; this situation is obtained by interchanging t and v in Figure 2. In this case define v as the last point on bd(P_{out}) that is visible to t, in a counterclockwise traversal of bd(P_{out}) that begins at t. Taking into account the two facts that t is responsible for watching the points mentioned above which are in the immediate neighborhood of r, and the set of all points visible to t forms a closed set, we can conclude that $v \in [q, r]$.

Taken together, these two cases imply that there is a guard t, with one of the following two properties:

$(a)\quad t \in [q,r) \quad \text{and} \quad v \in (r,q]$

$(b)\quad t \in [r,q] \quad \text{and} \quad v \in [q,r].$

By the definition of the points r and v, the two line segments $[q, r]$ and $[t, v]$ are guaranteed to be completely contained in the annulus. Further, both (a) and (b) guarantee that these two line segments intersect at some point inside the annulus (P_{out}–P_{in}); assume that this point is given by z, as shown in the figure. Then the line segments $[q, z]$ and $[z, t]$ form two consecutive edges that are also completely contained in the annulus; they will be regarded as the first two edges on this nested polygon. Proceeding further clockwise, to find the next guard after t, with the same property as above, we can continue the construction of this nested polygon, edge by edge. The maximum number of edges in nested polygon thus constructed will therefore be twice the number of guards in this solution, i.e., $2p$. Since $p < \lceil k / 2 \rceil$, $2p$ will be less than k. Hence we would have obtained a nested polygon with fewer than k edges, which contradicts the optimality of P^*. ∎

Lemma 1 also implies that:

Corollary 2. *Given a pair of nested polygons P_{in} and P_{out} that are not necessarily convex, if the optimal solution to Problem (P_{out}/P_{out}) has p guards, then $k \le 2p$; i.e., P^* has at most $2p$ edges.*

Assume now that both P_{in} and P_{out} are convex, as shown in Figure 1, and consider a solution that locates a guard on every odd vertex of P^*; thus assuming that x is the first vertex of P^* in the figure, this solution would place guards at x and z. We claim that this solution is feasible for Problem (P_{out}/P_{out}) and therefore can collectively watch entire bd(P_{out}). To prove this statement, suppose that it is not true and hence, there is some point $t \in \mathrm{bd}(P_{out})$, as shown in Figure 1, that is unwatched in this solution. This implies that this solution does not locate any guard in Out(t), which is given by the two consecutive segments $[i, t]$ and $[t, j]$ of $\mathrm{bd}(P_{out})$ in this case. However, Lemma 2 of Aggarwal et al.[2] dictates that each of the segments $[i, t]$ and $[t, j]$ of $\mathrm{bd}(P_{out})$ contains a vertex of P^*. If this implies that P^* has a vertex at t then our claim is true, since locating a guard at every odd vertex of P^* ensures that every vertex of P^* is watched. If however, P^* does not have a vertex at t, then there is one vertex of P^* in each of the segments $[i, t]$ and $[t, j]$ of $\mathrm{bd}(P_{out})$. By locating a guard at every odd vertex, our solution ensures that there is a guard on one of these vertices, thereby implying once again that t is watched in this solution. This enables us to state that:

Corollary 3. *Given a pair of convex nested polygons P_{in} and P_{out}, placing a guard at every odd vertex of P^* gives a solution to Problem (P_{out}/P_{out}) for this pair of polygons that is feasible. Further, since this solution has exactly $\lceil k / 2 \rceil$ guards, it is optimal too.*

2. Algorithms For Problems (P_{in}/P_{in})–(Ann/Ann)

In this section we will discuss all the problems listed in Table 1 individually, and give algorithms for solving them. As mentioned in the introduction, it will be assumed throughout that both P_{in} and P_{out} are convex polygons. Although there are nine different problems listed in the table, we show that it suffices to consider only a few of them.

2.1. Problems (P_{in}/P_{in}) and (P_{in}/Ann)

In Problem (P_{in}/P_{in}), the guards are required to watch only $\mathrm{bd}(P_{in})$. In Problem (P_{in}/Ann) however, they are required to watch the entire annulus. Nevertheless, as will now be shown, the two are equivalent.

Lemma 4. *Given a pair of convex nested polygons P_{in} and P_{out}, the optimal solutions to Problems (P_{in}/P_{in}) and (P_{in}/Ann) are identical.*

Proof. Since $\mathrm{bd}(P_{in})$ is a subset of the annulus (P_{out}–P_{in}), every feasible solution of Problem (P_{in}/Ann) is also feasible for Problem (P_{in}/P_{in}). To prove the converse, assume that it is not true. Hence, in some feasible solution to Problem (P_{in}/P_{in}), there is a point q in the annulus, that is not watched. Since q is not watched in this solution, there cannot be any guard placed In(q). That implies that the relative interior of In(q) is also unwatched in this solution, which makes it infeasible for Problem (P_{in}/P_{in}), too. Thus every feasible solution to Problem (P_{in}/P_{in}) is also feasible for Problem (P_{in}/Ann) and vice versa, and hence their optimal solutions will also be the same. ■

Since the optimal solution to Problems (P_{in}/P_{in}) and (P_{in}/Ann) can be assumed to be the same, we will only discuss Problem (P_{in}/P_{in}). It is easy to see the following: if the optimal solution to this problem locates a guard in the interior of an edge, say inside (w, r) in Figure 1, then this guard can be replaced by one at either of the vertices w or r without disturbing the optimality of the solution. This enables us to conclude that in the optimal solution of Problem (P_{in}/P_{in}), all guards are located on vertices of P_{in}. Since P_{in} is opaque, a guard located at a vertex, say r, can only watch the two adjacent edges $[w, r]$ and $[r, s]$, implying that at least $\lceil n / 2 \rceil$ guards are needed in any feasible solution. Consider the solution where a guard is placed at every odd vertex of P_{in}; this is a feasible solution to Problem (P_{in}/P_{in}), and furthermore, it has exactly $\lceil n / 2 \rceil$ guards. Hence, it is also optimal.

2.2. Problem (P_{in}/P_{out})

In this problem, the guards are required to watch $\mathrm{bd}(P_{out})$ while restricted to be only on $\mathrm{bd}(P_{in})$. Consider any optimal solution to Problem (P_{in}/P_{out}), and assume that there is a guard in this solution, that has been placed at a point x on $\mathrm{bd}(P_{in})$. If x is not a vertex of P_{in}, it must lie on an edge; assume that this edge is $[y, z]$. Then it is evident that this guard placement at x can be replaced by one at either y or z, without disturbing the optimality of the solution. This is because the set of points on $\mathrm{bd}(P_{out})$ that are visible to x is a proper subset of the set of points on $\mathrm{bd}(P_{out})$ that are visible to either one of y or z. That allows us to claim:

Lemma 5. *Given a pair of convex nested polygons P_{in} and P_{out}, the optimal solution to Problem (P_{in}/P_{out}) locates the guards only on vertices of P_{in}.*

Thus by Lemma 5, all guards in Problem (P_{in}/P_{out}) can be assumed to be located on vertices of P_{in}. For any vertex v of P_{in}, consider Out(v), which represents all points on $\mathrm{bd}(P_{out})$ that are visible to it. Due to the convexity of P_{in} and P_{out}, this entire segment is continuous; hence, by using the Circle Mapping Transformation to map P_{out} onto a circle, this segment can be mapped onto an arc on this circle.

Thus one way to solve Problem (P_{in}/P_{out}) is to reduce it to the Circle Covering Problem in the following manner. First find the segments Out(v) for each vertex v of P_{in}. Then by using the Circle Mapping Transformation to map P_{out} onto a circle, all these Outer Visibility Segments can be mapped onto arcs on this circle. Now, by solving the Circle Covering Problem for this circle, we can easily find the optimal placement of the guards from the arcs chosen in the minimal cover of this circle. That is the idea behind the following algorithm.

Algorithm—Problem (P_{in}/P_{out})

begin
{
1. Number all vertices of P_{in} from 1 to n clockwise.
2. **for** ($v = 1, \ldots, n$) **do**
   {
   Choose vertex v of P_{in}.
   Find the segment Out(v) on $\mathrm{bd}(P_{out})$.
   Label Out(v) as the segment corresponding to vertex v.
   $v = v + 1$.
   }
3. Apply Circle Mapping Transformation to map P_{out}

onto a circle, and all segments on bd(P_{out}) found in Step 2 above, to arcs on this circle.

4. Solve the Circle Covering Problem with the circle and arcs as given by Step 3.
5. The optimal placement of guards is then given by noting the arcs in the optimal solution found in Step 4 above, and locating guards only on those vertices of P_{in} that labeled the segments corresponding to these arcs.

}

end ■

Steps 1 and 2 in Algorithm—Problem (P_{in}/P_{out}) can be performed in $O(n + m)$ time to determine all the segments in a sorted order. Then the Circle Mapping Transformation of Step 3 can be done in $O(n + m)$ time to find all the arcs. As all arcs are given in a sorted order, the Circle Covering Problem in Step 4 can also be performed in $O(n + m)$ time, by using the algorithm in Chandru, Rajan, and Swaminathan.[7] Thus the entire algorithm takes $O(n + m)$ time.

It is worthwhile to note that it is also possible, albeit not as efficient, to solve Problem (P_{in}/P_{in}), and therefore Problem (P_{in}/Ann), by reducing it to the Circle Covering Problem. It has already been argued that the optimal solution to these problems will locate the guards only at the vertices of P_{in}. Thus we can first find the segments In(v) for each vertex v of P_{in}. Having done that, the Circle Mapping Transformation can then be used to map P_{in} onto a circle, and all these Inner Visibility Segments onto arcs on this circle. The solution to the Circle Covering Problem for this circle, will then give us the optimal solution to Problem (P_{in}/P_{in}).

2.3. Problem (P_{out}/P_{in}) and (P_{out}/Ann)

In both these problems, the guards are restricted to be on bd(P_{out}). However, in Problem (P_{out}/P_{in}) they are required to watch bd(P_{in}), whereas in Problem (P_{out}/Ann), they are required to watch the entire annulus. It will now be shown that despite this difference, the two problems are essentially equivalent.

Lemma 6. *Given a pair of convex nested polygons P_{in} and P_{out}, the optimal solutions to Problems (P_{out}/P_{in}) (P_{out}/Ann) are identical.*

Proof. Since bd(P_{in}) is a subset of the annulus (P_{out}–P_{in}), every feasible solution of Problem (P_{out}/Ann) is also feasible for Problem (P_{out}/P_{in}). To prove the converse, assume that it is not true. Hence, in some feasible solution to Problem (P_{out}/P_{in}), there is a point q in the annulus, that is not watched. Since q is not watched in this solution, there cannot be any guard placed anywhere in Out(q). But that implies that the relative interior of the segment In(q) is also unwatched in this solution, and hence, this solution is infeasible for Problem (P_{out}/P_{in}) too. Thus every feasible solution to Problem (P_{out}/P_{in}) is also feasible for Problem (P_{out}/Ann), and vice versa, and hence their optimal solutions will also be the same. ■

Since the optimal solution to Problems (P_{out}/P_{in}) and (P_{out}/Ann) can be assumed to be the same, we will only discuss Problem (P_{out}/P_{in}). In Problem (P_{out}/P_{in}), it is required to watch entire bd(P_{in}), while restricting the guards to be only on bd(P_{out}). Although these guards can be placed anywhere on bd(P_{out}), it will now be shown that only some points on bd(P_{out}) need to be considered for possible guard locations; these will be referred to as *critical points*. These critical points are the points of intersection obtained when each edge of P_{in} is extended to intersect with bd(P_{out}). For example, in Figure 3, the critical points produced in this manner are given by 1 through 8. With each edge of P_{in} producing two such critical points, there will be a total of $2n$ of them.

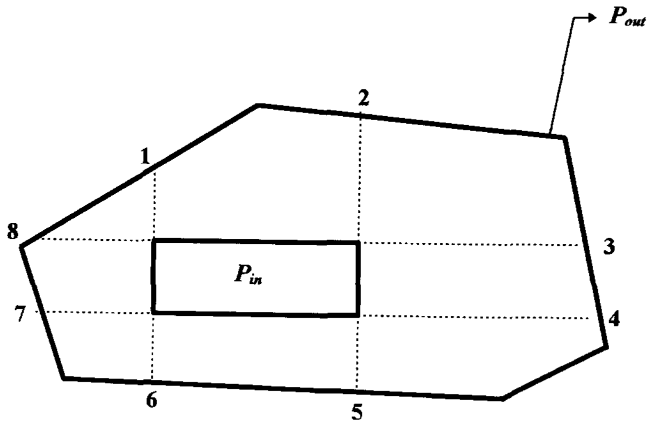

Figure 3. Critical points.

Lemma 7. *Given a pair of convex nested polygons P_{in} and P_{out}, in the optimal solution to Problem (P_{out}/P_{in}), all guards are located only on the critical points that have been found as above, by extending each edge of P_{in} to intersect with bd(P_{out}).*

Proof. Suppose not, and assume that in the optimal solution to Problem (P_{out}/P_{in}) there is a guard placed at a point x on bd(P_{out}), where x is not a critical point as defined above. Then x must lie between two adjacent critical points; let these be y and z respectively. It is evident that the set of points on bd(P_{in}) that are visible to x, is a proper subset of the set of points of bd(P_{in}) that are visible to either one of y or z. Therefore, replacing this guard at x with another that is located at either one of the points y or z still keeps this solution optimal. ■

Thus by Lemma 7, all guards in Problem (P_{out}/P_{in}) can be assumed to be located only on the critical points. For any such critical point v, consider In(v), which represents all points on bd(P_{in}) that are visible to it. Due to the convexity of P_{in} and P_{out}, this entire segment is continuous; hence, by using the Circle Mapping Transformation to map P_{in} onto a circle, this segment can be mapped onto an arc on this circle. Thus one way to solve Problem (P_{out}/P_{in}) is by reducing it to the Circle Covering Problem, along similar lines as Problem (P_{in}/P_{out}). In this case however, the polygon P_{in} is mapped onto a circle and the arcs on the circle are obtained from corresponding Inner Visibility Segments of the critical points. This is the basis for the algorithm given below for Problem (P_{out}/P_{in}).

Algorithm—Problem (P_{out}/P_{in})

begin
{
1. Extend all edges of P_{in} to intersect with bd(P_{out}) to get the *critical points*.
2. Number the critical points obtained in Step 1 from 1 to $2n$ clockwise.
3. **for** (v = 1, . . . , $2n$) **do**
   {
   Choose the vth. critical point.
   Find the segment In(v) on bd(P_{in}).
   Label In(v) as the segment corresponding to the critical point v.
   $v = v + 1$.
   }
4. Apply Circle Mapping Transformation to map P_{in} onto a circle, and all segments on bd(P_{in}) found in Step 3 above, to arcs on this circle.
5. Solve the Circle Covering Problem with the circle and arcs as given by Step 4.
6. The optimal placement of guards is then given by noting the arcs in the optimal solution found in Step 5 above, and locating guards only on those critical points that labeled the segments corresponding to these arcs.

}
end ■

Steps 1 and 2 of Algorithm—Problem (P_{out}/P_{in}) can be performed in $O(n + m)$ time, to get all the critical points in a sorted order. Since there are $O(n)$ critical points, and they all are given in a sorted order, Step 3 can be performed in $O(n + m)$ time to get the Inner Visibility Segments corresponding to these critical points. The remaining steps are similar to Algorithm—Problem (P_{in}/P_{out}) above and will also take $O(n + m)$ time. Thus it can be claimed that the entire algorithm takes $O(n + m)$ time.

2.4. Problem (P_{out}/P_{out})

In this problem, it is required to watch entire bd(P_{out}) while restricting the guards to be only on bd(P_{out}). By Corollary 3, if we find the Minimal Nested Polygon P^*, and then place a guard at every odd vertex of P^*, this placement will also be optimal for Problem (P_{out}/P_{out}). As mentioned before, due to the convexity of both P_{in} and P_{out}, it is guaranteed that P^* will always be convex in this case. Hence the algorithm from Aggarwal et al.[2] can be used to obtain the P^*, and thereby the solution to Problem (P_{out}/P_{out}) in $O((m + n)\log k)$ time.

It is also possible, albeit not as efficient, to solve it by reducing it to the Circle Covering Problem. Recall that the key step in the reduction is the identification of a polynomial number of discrete points as the only possible guard locations. It will be shown now that even for Problem (P_{out}/P_{out}), there exists such a set of points.

Lemma 8. *There are $O((m + n)k)$ points on* bd(P_{out}) *with the property that the optimal solution to Problem (P_{out}/P_{out}) places guards only at these points.*

Proof. It is known from Corollary 3, that the optimal solution to Problem (P_{out}/P_{out}) is obtained by finding a Minimal Nested Polygon P^* and placing a guard at every alternate vertex of it. Bhadury and Chandrasekaran[5] have shown that there is a set of $O((m + n)k)$ points on bd(P_{out}) with the property that there exists at least one Minimal Nested Polygon P^*, all of whose vertices can be assumed to belong to this set. These points are given as follows: i) All vertices of P_{in} and P_{out} and all the critical points found in Lemma 7; ii) All vertices of both clockwise and counterclockwise greedy structures obtained from every point in (i) and finally; iii) all the tight points on bd(P_{out}).

It is shown therein that the total number of these points is $O((m + n)k)$ and that these can be found in $O((m + n)k)$ time. Thus by the definition of these points it can also be assumed that the optimal solution to Problem (P_{out}/P_{out}) places guards only at points that belong to this set of $O((m + n)k)$ points. ■

2.5. Problems (Ann/P_{in}), (Ann/P_{out}) and (Ann/Ann)

These are the last three problems to be considered. We will now show that it is not necessary to devise separate solution procedures for them, because their optimal solutions are similar to those of Problems (P_{out}/P_{in}), (P_{out}/P_{out}), and (P_{out}/Ann) respectively.

Lemma 9. *Given a pair of convex nested polygons P_{in} and P_{out}, the optimal solutions to Problems (P_{out}/P_{in}), (P_{out}/P_{out}), and (P_{out}/Ann) are also optimal for Problems (Ann/P_{in}), (Ann/P_{out}), and (Ann/Ann) respectively.*

Proof. Assume that in an optimal solution to either Problem (Ann/P_{in}), (Ann/P_{out}), or (Ann/Ann), a guard has been placed at a point q inside the annulus (P_{out}–P_{in}). If q is not on bd(P_{out}), then choose a point r P_{in}, and let s be the point where a ray from r, in the direction of q, intersects with bd(P_{out}). Then it is evident that the set of all points on bd(P_{in}), bd(P_{out}), and (P_{out}–P_{in}) that are visible to q is a proper subset of all points on bd(P_{in}), bd(P_{out}), and (P_{out}–P_{in}) respectively, that are visible to s. Therefore, the guard placed at q can be replaced by one at s without disturbing the optimality of the solution. Thus, it can be assumed, without loss of generality, that in the optimal solutions to Problems (Ann/P_{in}), (Ann/P_{out}), and (Ann/Ann), all guards are placed only on bd(P_{out}). This makes the optimal solution to Problems (P_{out}/P_{in}), (P_{out}/P_{out}), and (P_{out}/Ann) optimal for Problems (Ann/P_{in}), (Ann/P_{out}), and (Ann/Ann), respectively. ■

3. Extensions

In this section we extend the results of the previous sections. First we show that the Art Gallery problems of this article are related to some well known concepts in Integer Programming. Then, we discuss how efficient parallel algorithms can be devised for them.

3.1. Art Gallery Problems and Integer Programming

All of Problems (P_{in}/P_{in}) through (Ann/Ann) are either solved by reduction to the Circle Covering Problem, or can be solved by this reduction. It is evident that the reason why this reduction works is because of the property that for any point in the annulus, both the Inner and Outer Visibility Segments are continuous. This enables us to map these segments onto arcs on a circle and thereby reduce these problems to an equivalent Circle Covering problem. This property is a direct result of the convexity of the two polygons P_{in} and P_{out}.

However, this property also makes it possible to formulate all of these problems as Integer Programs (IP). These IPs have an elegant property well known in the Integer Programming literature as the *Circular Ones* property, making it possible to solve all of the Problems in Table 1 using this IP formulation. This thus exposes the underlying mathematical structure of these problems and makes it possible to solve some extensions of these basic problems, which cannot be reduced directly to the Circle Covering Problem. Further, this also enables us to make an interesting conjecture about Art Gallery problems on spiral polygons.

Consider the general 0/1 IP:

$$\text{Min } \mathbf{e}^T X \tag{1}$$

$$\text{subject to } \quad \{AX \geq \mathbf{e};\ X\text{: } 0/1\}$$

where X is a $(n \times 1)$ vector, A is a $(m \times n)$ matrix all of whose entries are either 0 or 1, and $\mathbf{e}$ is the vector of all 1s whose dimensions are defined by context. It is well known (see, for example, Nemhauser and Wolsey[20]) that Eq. 1 is NP Hard. However, if the matrix A has some special properties then there are polynomial time algorithms available in the literature. One such property is that of *Circular Ones*.

Definition 10. A 0/1 matrix is said to have the Circular Ones property if in every row of this matrix, the 1s occur in consecutive columns, with the first and the last column being considered consecutive.

From Bartholdi, Orlin, and Ratliff,[4] it is known that if A has Circular Ones property, then Eq. 1 can be solved in polynomial time. The solution is obtained either by solving it by a linear programming-based algorithm or parametrically as a bounded series of Minimum Cost Network Flow Problems. By adopting the latter approach (which is also more efficient), and using the well known algorithm of Orlin[23] to solve the Minimum Cost Network Flow Problem, the above IP can be solved in $O(n \log m \log n(n + m \log m))$ time. In addition, if the constraint matrix A has this property, then more general versions of Eq. 1, where the objective function coefficients and right hand sides values are numbers other than 1, can also be solved in polynomial time. Furthermore, given a 0/1 matrix that does not have the Circular Ones property, Tucker[29] gives an $O(m^2 n)$ time algorithm to check if the columns of this matrix can be permuted so that the permuted matrix does have the Circular Ones property. If this is possible, the algorithm also gives this permutation.

In order to show how all the Problems (P_{in}/P_{in}) through (Ann/Ann) can be posed as Integer Programs of the form of Eq. 1, with constraint matrices that have the Circular Ones property, it will be helpful to explain the general idea behind these formulations. As a first step, we will identify a polynomial number of points with the property that the optimal solution can be assumed to place guards only at these points. A Boolean variable, i.e., a 0/1 variable, x_i will then be associated with each such point i, which will be defined as follows:

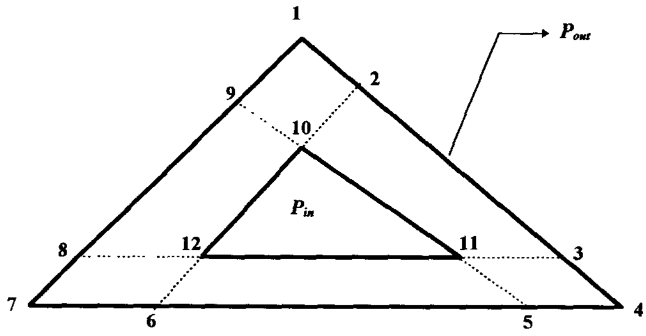

Figure 4. Example for IP formulations.

$$x_i = \begin{cases} 1 & \text{if a guard is placed at point } i \\ 0 & \text{otherwise} \end{cases}$$

The next step will be to consider each point i individually, and determine the entire area visible to a guard placed at i. Once this is done for all points, the entire area that is required to be watched would have been partitioned into regions, such that each region is visible from some of these points and (possibly) not from others. Then a *visibility matrix* A will be constructed, for which each entry a_{ji} will be determined as follows:

$$a_{ji} = \begin{cases} 1 & \text{if a guard placed at point } i \\ & \quad \text{can watch the entire region } j \\ 0 & \text{otherwise} \end{cases}$$

This *visibility matrix* will serve as the constraint matrix of the IP. There will be one constraint for each partition/region, that will ensure that in any feasible solution to the IP, the entire area corresponding to this partition is watched. The objective function of the IP will seek to minimize the number of guards necessary to satisfy all the constraints. Assuming that the polygons P_{in} and P_{out} are as given in Figure 4, we will now illustrate this formulation for Problem (P_{in}/P_{out}). The formulation for the other problems can be done similarly.

Problem (P_{in}/P_{out}).

By Lemma 5, the only guard positions that we need to consider are the vertices of P_{in}, which in this case are given by the points 10, 11, and 12 in Figure 4. Let the variables corresponding to these points be x_{10}, x_{11}, and x_{12}, respectively. The area to be watched is bd(P_{out}). By finding the Outer Visible Segment for each of the points 10, 11, and 12, it can be verified that these segments partition the entire boundary into the following intervals: [2, 3], [3, 5], [5, 6], [6, 8], [8, 9], [9, 2]. By checking which one of the vertices can watch each of these intervals, it is easy to verify that for this problem, the visibility matrix A is given by the following 0/1 matrix.

$$
\begin{array}{c}
\begin{array}{ccc} \mathbf{10} & \mathbf{11} & \mathbf{12} \end{array} \\
A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\begin{array}{l} \text{(for segment [2, 3])} \\ \text{(for segment [3, 5])} \\ \text{(for segment [5, 6])} \\ \text{(for segment [6, 8])} \\ \text{(for segment [8, 9])} \\ \text{(for segment [9, 2])} \end{array}
\end{array}
\qquad (2)
$$

To see how this visibility matrix above is obtained, consider, for example, the entire interval [2, 3]. This interval is visible from vertices 10 and 11, but not from 12; hence, the first row in the matrix A above is $(1 \quad 1 \quad 0)$. The other entries are obtained similarly. Then, on basis of the visibility matrix A in (2), the Integer Program to solve Problem (P_{in}/P_{out}) is given as:

$$\text{Min } x_{10} + x_{11} + x_{12}$$

subject to

$$
\begin{array}{lll}
x_{10} \phantom{+ x_{11}} + x_{11} & \geq 1 & \text{(for segment [2, 3])} \\
x_{10} + x_{11} + x_{12} & \geq 1 & \text{(for segment [3, 5])} \\
\phantom{x_{10} +} x_{11} + x_{12} & \geq 1 & \text{(for segment [5, 6])} \\
x_{10} + x_{11} + x_{12} & \geq 1 & \text{(for segment [6, 8])} \\
x_{10} \phantom{+ x_{11}} + x_{12} & \geq 1 & \text{(for segment [8, 9])} \\
x_{10} + x_{11} + x_{12} & \geq 1 & \text{(for segment [9, 2])} \\
x_{10}, x_{11}, x_{12}\text{: } 0/1 & &
\end{array}
\qquad (3)
$$

Since there is one variable for each vertex of P_{in}, there will be n variables in Eq. 3. Consequently there will be $O(n)$ Outer Visibility Segments and, therefore, as many constraints in Eq. 3. Furthermore, since these Outer Visibility Segments can all be found in polynomial time, it is guaranteed that Eq. 3 can be formulated in time that is polynomial in n and m, and that its size will also be polynomial in n.

To verify the correctness of the formulation, note that every feasible solution to Eq. 3 corresponds to a feasible placement of guards in Problem (P_{in}/P_{out}), as it ensures that the entire bd(P_{out}) is watched. The converse is also true since any placement of guards at vertices 10, 11, and 12 which ensures that entire bd(P_{out}) is watched, is also a feasible solution to Eq. 3. This ensures the correctness of the formulation in Eq. 3.

It is evident from Eq. 3 that the constraint matrix A, which is basically the visibility matrix for this problem, always has the Circular Ones property, if the vertices of P_{in} are given in the same order as they appear in the polygon. This is a direct result of the convexity of P_{in} and P_{out}, as it ensures that all those vertices of P_{in} that are visible to any one particular segment of bd(P_{out}) occur consecutively. Along with the fact that the size of Eq. 3 is polynomial in n, this enables this entire formulation to be completed and solved in time that is polynomial in n and m.

By reducing them to a corresponding Circle Covering Problem, it can be verified that all of the other problems can be solved similarly by formulating as IPs. This leads to:

Lemma 11. *All the Art Gallery problems stated in Table 1 can be formulated as 0/1 Integer Programs. Both the size of these IPs and the time taken to formulate them is polynomial in n and m. Furthermore, given that both P_{in} and P_{out} are convex, the constraint matrix in all of these formulations is guaranteed to have the Circular Ones property. These make it possible to be solve these IPs in polynomial time.*

The IP formulations above expose the underlying mathematical structure of these nine problems: the elegant property of Circular Ones in all of their Visibility Matrices. This exposition enables us to make the following conjecture about Art Gallery problems on spiral polygons. Everett and Corneil[11] have shown that the vertex visibility graph of spiral polygons is an interval graph, and MacDonald and Shermer[19] have strengthened this result by showing that the continuous graph of visibility of points of a spiral polygon is an (uncountably infinite) interval graph, and further, spiral polygons are the only ones with this property. One graph that is closely related to the interval graph is a circular arc graph, which in turn, is related to matrices with the Circular Ones property. On the basis of this, we conjecture that visibility matrices of Art Gallery problems on spiral polygons will have a property similar to the Circular Ones property, namely, the *Consecutive Ones* property (see Fulkerson and Gross[12]). If this is true, then by formulating them as IPs along similar lines as done in this article, these problems could also be solved in polynomial time (Nemhauser and Wolsey[20]).

Aside from the conjecture above, the exposition of this basic property also enables us to solve some extensions of these basic nine problems. Consider, for example, the following extension, which is similar to the Prison Yard Problem that was posed independently by D. Wood and J. Malkelvitch in [25], p. 146, and solved by Füredi and Kleitmann.[13]

Problem ($P_{nest}/P_{in}, P_{out}$) (*Prison Yard Problem for Convex Nested Polygons*).

Assume that in addition to P_{in} and P_{out}, we are also given a convex polygon P_{nest}, with l vertices, which is nested, i.e., completely contained in the annulus $(P_{out}–P_{in})$. Further, unlike P_{in}, the polygon P_{nest} is not opaque, i.e., a guard on the boundary of P_{nest} can watch bd(P_{in}). What is the minimum number of guards required to watch both bd(P_{in}) and bd(P_{out}) simultaneously if the guards are restricted to be only on the boundary of P_{nest}.

Thus, Problem $(P_{nest}/P_{in}, P_{out})$ is a combination of Problem (P_{in}/P_{out}) and (P_{out}/P_{in}). For example, in Figure 5 the two guards at x and y on the boundary of P_{nest} can watch both bd(P_{in}) and bd(P_{out}). By using arguments similar to the ones in Lemma 5 for Problem (P_{in}/P_{out}) and Lemma 7 for Problem (P_{out}/P_{in}), it can be seen that for Problem $(P_{nest}/P_{in}, P_{out})$, we need to consider only the following points for the placement of guards: the vertices of P_{nest} and the points of intersection obtained when the edges of P_{in} are extended

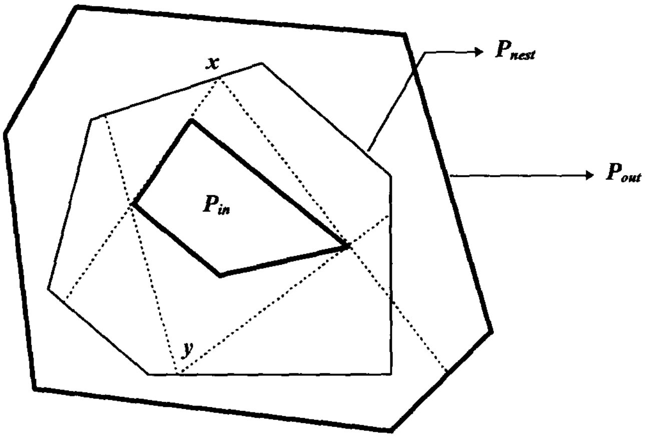

Figure 5. Prison Yard Problem for Convex Nested Polygons.

to meet with the boundary of P_{nest} (as done in Algorithm—Problem (P_{out}/P_{in})). Thus there will be $O(n + l)$ such possible guard locations. Then, by determining the segments of bd(P_{in}) and bd(P_{out}) that are visible to each of these possible guard locations, it is possible to formulate this as an IP of the form of Eq. 3. In this IP, there will be two sets of constraints: one corresponding to the segments on bd(P_{in}), to ensure that entire bd(P_{in}) is watched and another for segments on bd(P_{out}), to ensure the same for entire bd(P_{out}). As with Eq. 3, the size of this IP will be polynomial in n, m, and l, and the convexity of P_{in}, P_{out}, and P_{nest} will guarantee that its constraint matrix will have the Circular Ones property. Thus this IP can be solved in polynomial time, to obtain the solution to Problem (P_{nest}/P_{in}, P_{out}). On the other hand, an attempt to solve this problem by directly reducing it to the Circle Covering Problem, as was done for Problems (P_{in}/P_{in}) through (Ann/Ann), will fail, since there will be two circles to cover simultaneously: one for P_{in} and another for P_{out}.

Similar to Problem (P_{nest}/P_{in}, P_{out}), there are also other extensions of Problems (P_{in}/P_{in}) through (Ann/Ann) that can be solved easily by formulating as IPs, where the objective function or the right hand side coefficients are numbers other than one. For example, if there are different costs for locating guards at different points on the boundary of the two polygons, we will have a weighted version of these Art Gallery problems, where it may be required to find the "cheapest" placement of guards. That problem can be easily solved by formulating it as an IP of the form of Eq. 3, with the different costs being represented in the coefficients in the objective function. Since the constraint matrix is not affected by this, the resulting IP still has the Circular Ones property that makes it solvable in polynomial time.

3.2. Parallel Algorithms for Problems (P_{in}/P_{in})–(Ann/Ann)

We will now show that all of Problems (P_{in}/P_{in}) through (Ann/Ann) have efficient parallel algorithms. We design our parallel algorithms with respect to the well known Parallel Random Access Machines (PRAM). This is a synchronous model of computation, where each processor has access to the shared memory and all communications are performed through the shared memory. In this article, we allow several processors to read from the same memory location, but common write is disallowed. This model is known as the CREW-PRAM (Concurrent Read Exclusive Write PRAM). For details on the PRAM model of computation, see Karp and Ramachandran.[16] There is a vast body of literature on PRAM algorithms for problems in graph theory, computational geometry, numerical computation, etc. For example, we can sort n numbers in $O(\log n)$ time using $O(n)$ processors (Cole[10]). Similarly, the convex hull of n planar points can be computed in $O(\log n)$ time using $O(n)$ processors (Atallah and Goodrich[3]).

Rather than providing solutions for each of Problems (P_{in}/P_{in}) through (Ann/Ann), we extract the main computations and provide an efficient parallel implementation for each of them. The main computations are: i) Computing a mapping from the Art Gallery problem to the Circle Cover problem; ii) Computing an optimal circle cover; and iii) Computing a Minimum Nested Polygon for a given pair of nested convex polygons.

(i) can be performed as follows. Consider Problem (P_{in}/P_{out})—for each vertex $v \in P_{in}$, Out(v) is an arc on the circle corresponding to P_{out} (assuming that we have a clockwise orientation of the circle). To find the endpoints of the arcs, we need to know where the rays from vertex v intersect the boundary of P_{out}, i.e., bd(P_{out}). By performing a binary search based on polar coordinates, for each ray, we can find the points of intersection with bd(P_{out}) in $O(\log m)$ time. Hence in $O(\log m)$ time, using $O(n)$ CREW-PRAM processors, we can find all the arcs. Using the ideas from merging two sorted sequences of length n and m in $O(\log n + \log m)$ time using $O((n + m)/(\log n + \log m))$ CREW-PRAM processors (see Chandru et al[8]), we can compute all arcs optimally in parallel.

(ii) can be performed by adapting the parallel prefix to compute the optimal circle cover. Parallel prefix of n numbers can be performed in $O(\log n)$ time using $O(n/\log n)$ CREW-PRAM processors; see Karp and Ramachandran[16] and Jaja.[15]

(iii) can be performed in $O(\log n \log\log n)$ time using $O(n)$ CREW-PRAM processors by the algorithm of Chandru et al.[8]

Thus it can be concluded that we have an efficient parallelization for all the problems in Table 1.

4. Conclusions and Future Work

In this article we have studied a group of Art Gallery problems that are defined for a pair of convex nested polygons. We have shown that all of these can be solved in polynomial time, by either reducing them to the Circle Covering Problem, or by exploiting the properties of the Minimal Nested Polygon for these polygons. We also show that these problems can be solved in polynomial time by formulating them as Integer Programs. In all of these formulations, the constraint matrices have the desirable Circular Ones property. Finally, it is also demonstrated how efficient parallel algorithms may be devised to solve these Art Gallery Problems.

There are many different problems in this area that are open. Future research may examine the possibility of developing a geometric characterization of all polygons for which these Art Gallery problems can be solved by reduction to the Circle Covering Problem. Besides the Circular Ones property, there are also other properties of the constraint matrices for which Integer Programs can be solved in polynomial time. Another strand of future research may be to investigate if there are other types of polygons for which these Art Gallery problems can be formulated as Integer Programs with some of these other properties.

Acknowledgments

J. Bhadury was supported by UNB Grant #23-80 and NSERC Grant #OGP0121689. This support is gratefully acknowledged. The authors also wish to acknowledge the helpful comments and suggestions of S.N. Kabadi and two anonymous referees.

References

1. A. AGGARWAL, 1984. The Art Gallery Theorem: Its Variations, Applications and Algorithmic Aspects, Ph.D. Thesis, Johns Hopkins University.
2. A. AGGARWAL, H. BOOTH, J. O'ROURKE, S. SURI, and C.K. YAP, 1989. Finding Minimal Convex Nested Polygons, *Information and Computation 83,* 98–110.
3. M.J. ATALLAH and M.T. GOODRICH, 1986. Efficient Parallel Solutions to Some Geometric Problems, *Journal of Parallel and Distributed Processing 3,* 492–507.
4. J.J. BARTHOLDI III, J.B. ORLIN, and H.D. RATLIFF, 1980. Cyclic Scheduling Via Integer Programming with Circular Ones, *Operations Research 28,* 1074–1085.
5. J. BHADURY and R. CHANDRASEKARAN, 1996. Finding the Set of All Minimal Nested Convex Polygons, *Proceedings of Eighth Canadian Conference on Computational Geometry,* Ottawa, 26–31.
6. I. BJORLING-SACHS and D.L. SOUVAINE, 1991. A Tight Bound for Guarding Polygons with Holes, Technical Report LCSR-TR-165, Laboratory for Computer Science Research, Rutgers University, USA.
7. V. CHANDRU, V.T. RAJAN, and R. SWAMINATHAN, 1990. Monotone Pieces of Chains, Working Paper, IBM RC#16180.
8. V. CHANDRU, S. GHOSH, A. MAHESHWARI, V.T. RAJAN, and S. SALUJA, 1995. *NC*-Algorithms for Link Path Problems, *Journal of Algorithms* (to appear).
9. V. CHVATAL, 1975. A Combinatorial Theorem in Plane Geometry, *Journal of Combinatorial Theory Series B 18,* 39–41.
10. R. COLE, 1988. Parallel Merge Sort, *SIAM Journal on Computing 17,* 770–785.
11. H. EVERETT and D.G. CORNEIL, 1990. Recognizing Visibility Graphs of Spiral Polygons, *Journal of Algorithms 11,* 1–26.
12. D.R. FULKERSON and O.A. GROSS, 1965. Incidence Matrices and Interval Graphs, *Pacific Journal of Mathematics 15,* 835–855.
13. Z. FÜREDI and D. KLEITMANN, 1991. The Prison Yard Problem, *Combinatorica.*
14. F. HOFFMAN, M. KAUFMANN, and K. KRIEGEL, 1991. The Art Gallery Theorem for Polygons With Holes, *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science,* 39–48.
15. J. JAJA, 1993. *Introduction to Parallel Algorithms,* Addison-Wesley Reading, Massachussetts.
16. R. KARP and V. RAMACHANDRAN, 1990. Parallel Algorithms for Shared Memory Machines, in *Handbook of Theoretical Computer Science,* J. van Leeuwen, (ed.), North-Holland, Amsterdam, 869–941.
17. D.T. LEE and A.N. LIN, 1986. Computational Complexity of Art Gallery Problems, *IEEE Transactions on Information Theory,* IT-32, 276–282.
18. B.-C. LIAW, N. HUANG, and R. LEE, 1993. The Minimum Cooperative Guards Problem on *k*-Spiral Polygons, *Proceedings of 5th. Canadian Conference on Computational Geometry,* Waterloo, 97–102.
19. G. MACDONALD and T. SHERMER, 1994. Isomorphism of Spiral Polygons, *Discrete and Computational Geometry* (to appear).
20. G.L. NEMHAUSER and L.A. WOLSEY, 1988. *Integer and Combinatorial Optimization,* John Wiley and Sons, New York.
21. B.J. NILSSON, 1995. Guarding Art Galleries—Methods for Mobile Guards, Ph.D. Thesis, Lund University, Sweden.
22. B.J. NILSSON and D. WOOD, 1990. Optimum Watchmen Routes in Spiral Polygons, *Proceedings of 2nd. Canadian Conference on Computational Geometry,* 269–272.
23. J.B. ORLIN, 1988. A Faster Strongly Polynomial Minimum Cost Flow Algorithm, *Proceedings of the 20th. ACM Symposium on the Theory of Computing,* Chicago, 377–387.
24. F.P. PREPARATA and M.I. SHAMOS, 1989. *Computational Geometry: An Introduction,* Springer Verlag, New York.
25. J. O'ROURKE, 1987. *Art Gallery Theorems and Algorithms,* Oxford University Press, New York.
26. J. O'ROURKE and K.J. SUPPOWIT, 1983. Some NP-Hard Polygon Decomposition Problems, *IEEE Transactions on Information Theory IT-29,* 181–190.
27. T. SHERMER, 1984. Triangulation Graphs that Require Extra Guards, NYIT Computer Graphics Tech. Report, 3D–13.
28. T. SHERMER, 1992. Recent Results in Art Galleries, *Proceedings of IEEE 80,* 1384–1399.
29. A. TUCKER, 1971. Matrix Characterization of Circular Arc Graphs, *Pacific Journal of Mathematics 39,* 535–545.
30. S. VISHWANATHAN, 1993. The Edge Guard Problem for Spiral Polygons, *Proceedings of the 5th. Canadian Conference of Computational Geometry,* Waterloo, 103–108.
31. C.C. LEE and D.T. LEE, 1984. On a Circle-Cover Minimization Problem, *Information Processing Letters 18,* 109–115.